Call-A.P.P.L.E.™

Volume VI, Number 3 March 1983 $2.50 ($3.00 Canada, £1.75 UK.)

A CALL —A.P.P.L.E. REVIEW:

Anatomy of an Eamon Adventure

Reprinted with Permission of
Apple Pugetsound Program Library Exchange
www.callapple.org

Apple PugetSound Program Library Exchange

Anatomy of an Eamon Adventure

Robert Plamondon

How the EAMON Adventure System Works.

COMPUTERIZED adventure games have been surrounded by an aura of mystery that has nothing to do with their story lines. Simply stated, the mystery is this: no one knows how they work.

Like most mysteries, adventure programs are pretty straightforward once the hidden parts are revealed. To bring these secrets to light, let's look at the EAMON adventure system by Donald Brown, which is a set of BASIC programs for the APPLE][and available to members at $4.00 each on the A.P.P.L.E. "As Is" software series. EAMON is a good choice because it's a good system, it's all in BASIC, and it's in the public domain, which means that you can adapt or copy it at will. It does most of the things that all-text adventures do, so studying it can give you a good idea of how adventure games work.

The Structure of EAMON.

EAMON is a set of programs on disk which call each other as necessary, passing data through text files. The number of programs varies slightly depending on whether you have the original version or an altered one. In my own version I have merged several of the programs together, reducing the number of programs somewhat.

Warning: This gets a little intricate. If you don't care to know about the different programs, and just want to get the crucial stuff, skip this section.

The HELLO program runs when you boot the EAMON Master Disk. It loads an impressive picture of a dragon to the high-resolution graphics display to give you something to look at, then it runs a program called THE WONDERFUL WORLD OF EAMON.

THE WONDERFUL WORLD OF EAMON is the program that comes up with characters for you to play in the adventure. It prompts for your character's name, then tries to find an existing character of the same name in the file CHARACTERS. If it doesn't find him, it asks if you spelled if correctly (giving you a chance to try again). If you insist your spelling was correct, the program decides that you must be starting a new character and creates one for you, storing the data in the file CHARACTERS, and the name of your character and his record number to the file THE ADVENTURER. It also gives you a chance to read the instructions (Listing 1).

If the name you gave matches with an existing record in CHARACTERS, his name and record number are written to THE ADVENTURER. In both cases, the next thing the program does is to issue the command, "RUN MAIN HALL."

The MAIN HALL is where characters go to buy and sell weapons, buy armour and spells, and depart for adventures. The MAIN HALL reads the number stored in THE ADVENTURER, then reads in the character record of the same number from CHARACTERS.

After doing all the buying and selling that appeals to him, the player has two options: he can send his character on an adventure, or quit. If he quits, his updated character record is resaved in CHARACTERS. If he decides to go on an adventure, the character record is partially erased (although the data still exists in program variables), and the player is prompted to insert the disk with the adventure on it. The character's data is written to a file named FRESH MEAT, and the file EAMON.NAME is read. EAMON.NAME contains the name of the adventure program. The MAIN HALL then runs the program of that name, and the main adventure begins.

Okay, so it's complicated. The code that does all the file manipulations is a lot shorter than the explanations, though, and passing parameters between program is not the crucial part of EAMON, anyway. We'll get to the good stuff in a minute.

The reason why the character record is partially erased before the character embarks on an adventure is to make sure that if he dies on the adventure, he stays dead. As it is, there are only two ways to get your character's statistics back into the CHARACTERS file. One is to survive the adventure. The other is to use the RESURRECT program on the EAMON Utility Disk (which contains other interesting programs, as well). Resurrecting a character is considered cheating by purists, but everyone does it anyway.

The program called by the MAIN HALL usually prints out introductory material for the adventure; telling you what's going on, what your character is supposed to do, and so forth. This program in turn calls *another* program (don't worry, we're almost there): the BASE DUNGEON PROGRAM, which does the actual adventure. The rest of this article will deal with the BASE program.

A (Nearly) Infinite Loop.

The main activity of an adventure program is a loop: the program gets a command from the player, figures out what it means, goes to the appropriate routines to carry out the command, and returns to get another command. This loop continues until the adventure is completed, the character dies, or the power is turned off.

EAMON ADVENTURE

Getting commands is an easy job. All it takes is a statement like "INPUT A$." Breaking the command into words, and identifying the actions corresponding to the words is also pretty easy, it turns out. The routines that carry out the actions are usually fairly simple, too.

So where's the hard part? The individual parts are all easy, but there are a *lot* of individual parts. A program that puts them all together can be immense. This makes the design job difficult enough that most people who'd like to write adventures never get started. Fortunately, EAMON makes a good base for all sorts of adventures, and the design work for the programs has already been done.

One Program, Many Adventures.

One of the most useful things about the EAMON system is that the same BASE program is used (with minor alterations) with all the EAMON adventures. This is possible because all the information about a specific adventure is kept in text files, and this data is what controls the adventure.

For example, the information about which rooms connect where is kept in a text file, which is loaded into an array at the beginning of the adventure. This data controls where the adventurer can go. Monster data is likewise kept in files, so the creatures you encounter will vary from one adventure to the next.

The files for each EAMON adventure are:

1) EAMON.DESC, which contains the long descriptions for each room, artifact, special effect, and monster in the adventure.
2) EAMON.ARTIFACTS holds the data on artifacts (i.e., objects) in the adventure, including names, values, initial placement, etc.
3) EAMON.MONSTERS contains monster data, and is similar to EAMON.ARTIFACTS.
4) FRESH MEAT holds information on the character.
5) EAMON.ROOMS holds the information on which room connects where.
6) EAMON. ROOM NAMES holds the names of each room.

These files provide all the information required to run a unique adventure.

The Program Listing.

Reading the listing of the EAMON adventure system is necessary if you want to understand the way EAMON works. I have added a large number of remarks to the listing to make it more comprehensible (there are very few comments in the original program.) Once you understand how this program works, you should be able to write your own adventure program, if that appeals to you. If you want to write adventures, but don't want to design a new program, EAMON is the system for you. It's only available for the Apple][(as far as I know), but it would be fairly easy to convert it to run on other machines.

This is my version of the BASE program, which is a modification of John Nelson's modification to Donald Brown's original. John Nelson's version is the one on the Dungeon Designer Version 5.0; mine is Version 5.1, and previous versions are (as far as I can tell) Donald Brown's original creation. A number of routines have been changed. Some of these make the program run faster, some correct errors in coding or concept in the original, and some make the output more readable.

A Brief Overview of the Program.

Lines 100-900 make up the main loop. Lines 100-200 print out a description for the room and each monster and object in it. If something has been encountered before, its name is printed. If something is being seen for the first time, the long description is printed.

Lines 200-290 get the player's command and process it. The input is broken into a subject and verb (S$ and V$), and the verb is compared to the list of commands in the string array C$. If a match is not found, an error message is printed along with a list of acceptable verbs. If there is a match, the appropriate routine is jumped to in line 290. If the player just hits RETURN, rather than typing anything, his previous command is repeated.

Lines 300-900 control combat. Each monster in the room gets its chance to attack. Morale is checked for each monster: if the morale check fails, the monster flees from the room. A monster's morale decreases as it is wounded. Line 900 is "GOTO 100" — forming the (nearly) infinite loop.

Lines 1000–2000 handle initialization. Variable and arrays are set to their initial values, and data is read in from disk. The list of command words is in line 1920.

Lines 2000–3000 handle the end of the game. If your character lives, he gets to sell his treasures and return to the MAIN HALL program. If he died, the MAIN HALL program runs THE WONDERFUL WORLD OF EAMON, giving you the chance to start a new character.

Lines 3000–4000 handle movement. The program determines which direction you want to move, and checks to see if it's possible. If it is, you are moved into another room.

Lines 3600–3900 handle monster reaction. If the monster hasn't been met before, "dice" — random numbers — are rolled and compared to its "Friendliness." The monster will be either friendly, neutral, or hostile. If may follow you from room to room (it will always do so if friendly, never if neutral, and will sometimes chase you when hostile). This makes monster reaction important to the movement routine.

Lines 4000–5000 handle picking things up. It checks to see that the item is in the room, and isn't too heavy. If the character is unarmed, and the item is a weapon, the weapon becomes his "ready" weapon — the one he is prepared to fight with. The command "GET ALL" causes the character to pick up everything that's not too heavy for him.

Lines 5000–6000 make up the DROP routine, which is the GET routine in reverse. DROP is used to get rid of unwanted baggage, and is helpful in combat to arm a comrade who has lost or broken his weapon. If a weapon is on the floor, any unarmed combatant will pick it up. (GIVE doesn't work during combat, though perhaps it should.)

Lines 6000–7000 are the LOOK or EXAMINE command, which is usually used in the form of "EXAMINE SWORD." The long description in EAMON.DESC is printed if the thing you're looking at is in the room.

Lines 7000–8000 make up the combat routine. The combat system is suspiciously similar to that in *RuneQuest,* a popular fantasy role-playing game (not a computer game). Each creature has a chance to hit with each category of weapon (axe, bow, spear, and sword). These numbers are in the range of zero to something over 100. If a random number between 1 and 100 is less than the chance to hit, the creature hit its opponent.

Critical Hits do extra damage, and *Fumbles* cause the attacker to drop his weapon or hurt himself.

Armor absorbs damage, but reduces the wearer's chance to hit because of its weight and bulk. *Armour Expertise* is the skill of compensating for the hampering effects of wearing armour.

If a creature scores a hit, there is a chance that his skill at the weapon he is using will increase (practice makes perfect).

Lines 7700–7900 handle dying creatures. When a creature dies, its weapons and possessions fall to the floor, and his dead body (an artifact) is placed in the room. Its Courage is reduced to 75% of its current level in case it is resurrected by a POWER spell (having just died, it would think twice before risking its life again). If the player's character dies, the game is over (this is one way out of the nearly infinite loop).

Lines 8000–9000 handle the FLEE command. The character flees through a randomly selected exit. Sometimes your enemies follow, and sometimes they don't.

Lines 9000–10000 are the GIVE routine. You can give anything you are carrying to a monster. This sometimes makes them friendlier. If you use a number, as in "GIVE ORC 500" the program assumes you are giving that many gold pieces to the monster. Money makes a good bribe.

Lines 10000–11000 form the INVENTORY routine, which prints out the name of everything your character is carrying.

Lines 11000–12000 form the BLAST spell, which is used to wound opponents. All spells in EAMON decrease in effectiveness with use: if you have an 80% chance of success on the first attempt to use the spell, it is reduced to 40% on the second try, 20% on the third, and so on. This is not a very effective way of limiting spell use, but no one has changed it yet.

Lines 12000–13000 form the HEAL spell, which cures wounds on your character, but can't be used to heal his friends (someone ought to fix this).

Lines 13000–14000 are the POWER spell, which does unpredictable things — from causing things to vanish to raising the dead to collapsing the roof on your head. Many adventures have a BASE DUNGEON PROGRAM with a modified POWER spell, just for variety.

Lines 14000–15000 make up the SPEED spell, which increases agility and the base chance to hit. SPEED is nice because you can cast it on yourself before a battle (when spell-casting is safe), and it lasts a reasonably long time.

Lines 15000–16000 are the SMILE command, which is a way to try to make friends. It doesn't really have any effect on the creatures, but it does tell you how they feel about you.

Lines 16000–17000 make up the SAY command, which give you an alternate way of casting spells, and little else. Some adventures use magic words, so this routine is modified to let you open doors by typing "SAY OPEN SESAME."

Lines 17000–18000 let you READY a weapon. You can carry a huge number of weapons, but you can only fight with one at a time. READYing a weapon is how you specify which one you're using.

Lines 18000–19000 are the "Save Game" routine, added to the program by John Nelson. The pointers to the start and end of variable and array space are saved into a binary file, and then the variable and array spaces are saved into two more binary files.

Lines 19000–20000 restore a saved game. By retrieving the variables and variable pointers, the game begins exactly where it left off.

Lines 50000–60000 are the error-handling routines, consisting of a short machine-language program that helps fix APPLESOFT's ONERR GOTO bugs, a routine to print out the error message (which is disabled by ONERR GOTO), and a jump back to 100. This doesn't always work, but it's helpful. If you are running an EAMON adventure that doesn't have the error-trapping routine, you can get back to where you left off by typing

```
POKE 51,0 : GOTO 100
```

The Base Program is shown in Listing 2. Table 1 gives an explanation of the variables in the program.

The basic instructions for playing EAMON are given in Listing 1, and tell a lot about the game, especially the combat details.

The Dungeon Designer Disk contains a number of programs that help you to create adventures. The most important of these, DUNGEON EDIT, is used to create the descriptions of rooms, monsters, and artifacts. There are several other useful programs, as well as the *Players' Manual* and the *Dungeon Designers' Manual* (which come on text files, with a program to print them out to the screen or a printer).

Getting EAMON. There are two good sources of EAMON adventures. One is good old *A.P.P.L.E.;* the other is the Apple Avocation Alliance, Inc.

A.P.P.L.E. has a number of adventures available on its "As Is Software" disks, which cost $4.00 each.

The Apple Avocation Alliance is in business to do essentially the same job as *A.P.P.L.E.'s* "As Is Software" — the distribution of public-domain software at low prices. Ron Maleika runs 3A almost single-handedly, and although he has complained about being swamped with orders recently, service has consistently been very fast — my orders have been filled from six days to two weeks from the time of mailing them in. *A.P.P.L.E.* averages about four weeks.

3A will ship disks on DOS 3.2 or 3.3, and Ron is actively acquiring all the EAMON adventures in the known universe. Not a bad deal. Table 2 gives a list of the EAMON adventures in 3A's October, 1982 catalog, plus two more that Ron told me about later.

If you are not an *A.P.P.L.E.* member, you may order direct from:

Apple Avocation Alliance, INC.
721 Pike Street
Cheyenne, Wyoming USA 82009

and enclose $2.00 for a catalog, and $3.00 for year's subscription. 3A has a huge number of public-domain programs which they sell for $1.00 plus the price of the disk (currently at $2.15). 3A has over thirty EAMON adventures, including the *EAMON Master Disk* (which you *must* have to run the adventures), the *Dungeon Designer Disk,* and two *EAMON Utility Disks.*

Conclusions.

A lot of the features in EAMON are inelegant, and some are downright crude. Still, EAMON makes a very good system for adventure games, especially combat-oriented adventures. Since it's in the public domain, anyone can use it and try to improve it — but in its present form it is eminently playable and very enjoyable.

The BASE PROGRAM shows you one way to write an adventure game, and makes a good starting point for those who want to write their own adventure systems. People tend to start out by just playing the adventures, but eventually they can't resist the temptation to write one.

The EAMON Adventures:

01	EAMON Master Disk and The Beginners' Cave
02	Minotaur's Lair
03	Caves of the Mind
04	Zyphur River Venture
05	Doom Castle
06	The Death Star
07	The Devil's Tomb
08	The Abductor's Quarters
09	Assault on the Clone Master
10	The Magic Kingdom
11	Molinar's Tomb
12	The Quest for Trezore
13	Treasure Island
14	Furioso
15	Heroes Castle
16	The Caves of Mondamen
17	Merlin's Castle
18	Hogarth Castle
19	The Death Trap
20	The Black Death
21	Marron Quest
22	The Senator's Chambers
23	The Temple of Ngurct
24	Black's Mountain
25	Nuclear Nightmare
26	Moleman Assault
27	Moleman Revenge
28	London Tower
29	The Lost Island of Apple
30	The Underground City
31	Gauntlet
32	House of Ill Repute
33	Nobbin's Hell Hole

EAMON Tournament Adventures:

60	Castle of Count Fuey
61	Search for the Key
62	The Rescue Mission

Utilities:

Dungeon Designers' Disk
EAMON Utilities I
EAMON Utilities II

Before they know it, they're writing their own adventure system.

Whether you're one of the fanatics described above, or just like to play adventure games, EAMON is worth looking into. The price is right, the variety is stunning, and the quality is surprisingly high.

Happy adventuring!

Table I
Variables used in EAMON's
BASE DUNGEON PROGRAM
from A Manual for EAMON Adventure Designers
by Donald Brown

AC -	ARMOUR CLASS of player character (how many points of damage the character's armour will absorb.
AD%(*,*) -	ARTIFACT DATA The first subscript is the number of the artifact, and the key for the seconds is (1 = Value, 2 = Type, 3 = Weight, 4 = Room, 5 = Complexity, 6 = Type, 7 = Dice, 8 = Sides, 9 = Flag is artifact has been seen before)
AE -	ARMOUR EXPERTISE
AN$(*) -	NAMES OF ARTIFACTS
BANK -	GOLD PLAYER HAS IN BANK
C -	HOLDS NUMBER OF COMMAND GIVEN
C$(*) -	VERBS PROGRAM RESPONDS TO
CH -	PLAYER CHARISMA
CA$ -	LAST COMMAND GIVEN
DF -	DEFENDER
DIE -	LOGICAL FLAG, 1 = PLAYER HAS DIED
DK$ -	HOLDS CTRL-D FOR DISK COMMANDS
DR%(*) -	ROOM MOVED IN EACH DIRECTION
EA -	EFFECT OF ARMOUR ON ODDS-TO-HIT
FD%(*) -	FULL DAMAGE OF A SIDE IN COMBAT
FR -	FUMBLE ROLL or FRIEND RATING
GOLD -	GOLD PLAYER HAS ON PERSON
HIT -	LOGICAL FLAG IF HIT IN COMBAT
INC -	LOGICAL FLAG IF ABILITY INCREASED
LK -	LOGICAL FLAG IF "LOOKED" ALREADY
MD%(*,*) -	MONSTER DATA First subscript is monster number, second key is (1 = HD, 2 = AG, 3 = FRIEND, 4 = COUR., 5 = ROOM, 6 = WGHT, 7 = DEF ODDS, 8 = ARMOUR, 9 = WEAPON #, 10 = ODDS TO HIT, 11 = W DICE, 12 = W SIDES, 13 = HITS TAKEN, 14 = REACTION; 0 = NOT MET, 1 = HOSTILE, 2 = NEUTRAL, 3 = FRIENDLY)
MN$(*) -	NAME OF MONSTER
MR -	MONSTER MORALE
NA -	NUMBER OF ARTIFACTS
NBTL -	LOGICAL FLAG IF IN BATTLE
NC -	NUMBER OF COMMANDS
NM :	NUMBER OF MONSTERS
NW -	TOTAL COUNT OF WEAPONS IN GAME
NZ :	NUMBER ARTIFACTS BESIDES THE PLAYER'S WEAPONS
LOF -	NUMBER OF OFFENSIVE (ATTACKING) MONSTER
RAISE -	LOGICAL FLAG IF POWER RAISED SOMEONE FROM THE DEAD
REC -	PLAYER RECORD IN CHARACTERS FILE
RL -	RANDOM NUMBER 1-100
ROOM -	ROOM PLAYER IS IN
RR -	RANDOM NUMBER 1-100 FOR POWER SPELL
S$ -	SUBJECT OF COMMAND GIVEN
S2%(*) -	CURRENT SPELL ABILITY
SA%(*) -	TOTAL SPELL ABILITY
SEX$ -	HOLDS "M" OR "F" FOR PLAYER
SPD -	NUMBER OF TURNS SPEED SPELL HAS TO GO
SUC -	LOGICAL FLAG IF SPELL SUCCEEDED
TD%(*) -	DAMAGE TAKEN IN BATTLE FOR SIDE
TP -	TOTAL PRICE OF TREASURE
V$ -	VERB OF COMMAND
V%(*) -	FLAGS IF PLAYER HAS BEEN IN ROOM
WA%(*) -	PLAYER'S WEAPON ABILITY
WD%(*) -	FOR WEAPON, DICE OF DAMAGE
WN%(*) -	NAME OF PLAYER'S WEAPON
WO%(*) -	WEAPON COMPLEXITY
WP%(*) -	WEAPON POINTER (USED AT THE END OF GAME)
WS%(*) -	SIDES/DIE OF DAMAGE FOR WEAPON
WT -	WEIGHT PLAYER CARRYING
WT%(*) -	WEAPON TYPE
WZ -	NUMBER OF WEAPONS PLAYER BROUGHT

Listing 1

You read the instructions and they say —

Information About THE WORLD OF EAMON

You will have to buy a weapon. Your chance to hit with it will be the weapon complexity, plus your ability in that class, plus twice your agility.

The five classes of weapons (and your current abilities with each) are —

Club/Mace	20%
Spear	10%
Axe	5%
Sword	0%
Bow	-10%

Every time you score a hit in battle, your ability in the weapon class may go up by 2%, if a random number from 1-100 is less than your chance to miss!

There are four armour types, and you may also carry a shield if you do not use a two-handed weapon. These protections will absorb hits placed upon you (almost always!) but they lower your chance to hit. The protections are —

Armour	Hits Protect	Odds Adjust
None	0	- 0%
Leather	1	-10%
Chain	2	-20%
Plate	5	-60%
Shield	1	- 5%

You will develop an armour expertise, which will go up when you hit a blow wearing armour and your expertise is less than the armour you are wearing. No matter how high your armour expertise is, however, the net effect of armour will never increase your chance to hit.

You can carry weights up to ten times your hardiness, or 150 gronds. (A measure of weight, one grond = 10 DOS.)

Additionally, your hardiness tells how many points of damage you can survive. Therefore, you can be hit with 15 1-point blows before you die.

However, you will not be told how many blows you have taken. You will be merely told things such as —

"Wow! That one hurt!"
or "You don't feel very well."

Your charisma (20) affects how citizens of EAMON react to you. You affect a monster's friendliness rating by your charisma less ten, difference times two (20%).

You start off with 200 gold pieces, which you will want to spend on supplies for your first adventure. You will get a lower price for items if your charisma is high.

After you begin to accumulate wealth, you may want to put some of your money into the bank, where it cannot be stolen. However, it is a good idea to carry some gold with you for use in bargaining and ransom situations.

You may also hire a wizard to teach you some magic spells. There are four spells you may learn.

Blast	—Hurt your enemies from a distance.
Heal	—Remove damage from your body.
Speed	—Double your dexterity for a time.
Power	—Does something weird. The exact effect is unpredictable.

Other types of magic may work in various adventures and items may have special properties. However, these will not work in other adventures than where they were found. Thus, it is best (and you have no choice but to) sell all items found in adventures, except for weapons and armour.

The man behind the desk takes back the instructions and says, "It is now time for you to start your life." He makes an odd sign with his hand and says, "Live long and prosper."

You now wander into the main hall.

```
]LIST

5  REM

        EAMON ADVENTURE
       BASE PROGRAM  2.1

        BY DONALD BROWN

    WITH MODIFICATIONS  BY
JOHN NELSON & ROBERT PLAMONDON

CALL -A.P.P.L.E. : MARCH 1983

10  ONERR  GOTO 50
15  REM
*** IF 'GAME.PTRS' EXISTS
     . A GAME HAS BEEN SAVED.
16  REM *** IF IT DOESN'T EXIST,
     THE ERROR WILL CAUSE A JUMP
     TO 50.
20 DK$ =  CHR$ (4): PRINT DK$"VER
     IFY GAME.PTRS": GOTO 19000
50  POKE 216,0: GOTO 1000: REM **
     * SET UP THE GAME.
100  REM
**** MAIN LOOP. GIVE RO
     OM DESC, THEN GET COMMAND.

105  SPEED= 255
110  PRINT
120  IF SPD THEN SPD = SPD - 1: IF
      NOT SPD THEN MD%(0,2) = MD%
     (0,2) / 2:MD%(0,10) = MD%(0,
     10) - 2 * MD%(0,2): IF  RND
     (1) > .8 THEN  PRINT "YOUR S
     PEED SPELL HAS JUST EXPIRED!
     ": PRINT
128  REM
*** PRINT ROOM DESCRIPTI
     ON. IF ROOM HASN'T BEEN SEEN
      BEFORE,
129  REM  *** GIVE A DESCRIPTION.
      OTHERWISE JUST PRINT THE RO
     OM NAME.
130  IF V%(ROOM) THEN  PRINT "YOU
      ARE": PRINT " ";RN$(ROOM): GOTO
     150
140  PRINT DK$;"READ EAMON.DESC,R
     ";ROOM: INPUT A$: PRINT DK$:
      PRINT A$: PRINT :V%(ROOM) =
     1
149  REM
*** PRINT OUT NAMES OR
     DESCRIPTIONS OF MONSTERS IN
     ROOM.
150  FOR M = 1 TO NM: IF MD%(M,5)
      <  > ROOM THEN  NEXT : GOTO
     170
155  IF MD%(M,15) THEN  PRINT "
      >> ";MN$(M);" IS HERE.": NEXT
      : GOTO 170
160  PRINT DK$;"READ EAMON.DESC,R
     ";M + 300: INPUT A$: PRINT D
     K$:MD%(M,15) = 1: IF A$ <  >
     "" THEN  PRINT A$: PRINT : NEXT

169  REM
*** PRINT OUT NAMES OR
     DESCRIPTIONS OF ARTIFACTS IN
      ROOM.
170  FOR A = 1 TO NZ: IF AD%(A,4)
      <  > ROOM THEN  NEXT : GOTO
     190
```

```
175  IF AD%(A.9) THEN  PRINT "
     -- YOU SEE ";AN$(A): NEXT : GOTO
     190
180  PRINT DK$;"READ EAMON.DESC.R
     ";A + 100: INPUT A$: PRINT D
     K$: PRINT A$:AD%(A.9) = 1: PRINT
     : NEXT
189  REM
*** PRINT OUT NAMES OF
     PLAYER CHARACTER'S DROPPED W
     EAPONS.
190  FOR A = A TO NA: IF AD%(A.4)
      = ROOM THEN  PRINT "   -> Y
     OUR ";AN$(A);" IS HERE."
200  NEXT A:V%(ROOM) = 1
205  REM
***** GET COMMAND *****


210  INVERSE : PRINT "YOUR COMMAN
     D?";: NORMAL : INPUT " ";A$
218  REM
*** PARSE INPUT:  GET R
     ID OF LEADING SPACES.
220  IF  LEFT$ (A$.1) = " " THEN
     A$ =  MID$ (A$.2): GOTO 220
229  REM
*** IF INPUT IS A NULL
     STRING. DO PREVIOUS COMMAND
     AGAIN.
230  IF A$ = "" THEN A$ = CZ$: VTAB
      PEEK (37): HTAB 17: PRINT A
     $: GOTO 275
239  REM
*** FIND END OF FIRST W
     ORD (THE VERB) BY LOOKING FO
     R A SPACE
240 CZ$ = A$: FOR A = 2 TO  LEN (
     A$): IF  MID$ (A$.A.1) <  >
     " " THEN  NEXT
249  REM
*** FIRST WORD IS VERB
     (V$). SECOND WORD IS SUBJECT
      (S$).
250 V$ =  LEFT$ (A$.A - 1):S$ =  MID$
     .(A$.A + 1)
255 LV =  LEN (V$)
260  IF  LEFT$ (S$.1) = " " THEN
     S$ =  MID$ (S$.2)  GOTO 260
269  REM
**** SEARCH THROUGH LIS
     T OF VERBS FOR MATCH WITH CO
     MMAND ***

270  FOR C = 1 TO 6: IF  LEFT$ (C
     $(C).LV) = V$ THEN 290
272  NEXT :F = 0: FOR A = 7 TO NC
       IF C$(A) = V$ THEN C = A: GOTO
     290
273  IF  LEFT$ (C$(A).LV) = V$ THEN
     F = F + 1:C = A: IF F > 1 THEN
     275
274  NEXT  IF F THEN 290
275  PRINT : PRINT "HUH? I ONLY U
     NDERSTAND THESE COMMANDS--
        ";: FOR C = 1 TO NC: PRINT
     C$(C); SPC( 20 -  LEN (C$(C)
     ));: NEXT : PRINT : PRINT : GOTO
     210
280  IF C$(C) = S$ THEN S$ = V$:V
     $ = C$(C)
285  SPEED= 200
289  REM
***** JUMP TO ROUTINE T
     HAT HANDLES COMMAND *****
```

```
290  ON C GOTO 3000.3000,3000.300
     0.3000.3000.4000.4000.5000.6
     000.6000.7000.8000,9000,1000
     0.11000.12000.13000.14000.15
     000.15000,16000.17000,18000
300  REM
***** COMMANDS RETURN T
     O HERE:  COMBAT LOOP *****

309  REM
*** IF NOT IN BATTLE. G
     OTO 500
310  IF  NOT NBTL THEN 500
319  REM
*** DO COMBAT FOR EACH
     MONSTER IN ROOM.
320  FOR M = 1 TO NM: IF MD%(M.5)
      <  > ROOM THEN 490
329  REM
*** MORALE CHECK.  SEE
     IF MONSTER TURNS TAIL AND RU
     NS.
330 M2 = MD%(M.14) - (MD%(M.14) =
     2):MR = 100 * FD%(M2) / TD%(
     M2) +  INT (41 *  RND (1) -
     20) - 20 * (MD%(M.9) =  - 1)

340  IF MD%(M.4) < MR THEN  INVERSE
     : PRINT MN$(M);" FLEES OUT A
     N EXIT.": NORMAL : GOSUB 850
     0:MD%(M.5) = R2:M2 = MD%(M.1
     4):TD%(M2) = TD%(M2) - MD%(M
     .1):FD%(M0) = FD%(M2) - MD%(
     M.13)  GOTO 490
344  REM
*** BRANCH ACCORDING TO
     MONSTER REACTION: 1=HOSTILE.
      2=NEUTRAL, 3=FRIENDLY ***
345  ON MD%(M.14) GOTO 360,490.39
     0
359  REM
*** MONSTER IS HOSTILE.
       HAVE HIM ATTACK SOMEONE ON
      OUR SIDE ***

360 DF = M  IF TD%(3) = MD%(0.1) THEN
     DF = 0: GOSUB 7500: GOTO 490

370  FOR M2 = 1 TO NM: IF MD%(M2.
     5) = ROOM AND MD%(M2.14) = 3
      AND  RND (1) < .25 THEN DF =
     M2: GOSUB 7500:M2 = 200: NEXT
     M2  GOTO 490
380  NEXT M2:DF = 0: GOSUB 7500: GOTO
     490
389  REM
*** MONSTER IS FRIENDLY
     .  HAVE HIM ATTACK AN ENEMY
     ***


390 DF = M
400  FOR M2 = 1 TO NM: IF MD%(M2.
     5) = ROOM AND MD%(M2.14) = 1
      THEN DF = M2: GOSUB 7500:M2
      = 200: NEXT M2: GOTO 490
410  NEXT M2
489  REM
*** IF ANY FOES ARE LEF
     T ALIVE. WE'RE STILL IN BATT
     LE.
490 NBTL = (FD%(1) < TD%(1)): IF
     NBTL THEN  NEXT M
500  REM
***** LOOP TO LINE 100
     FOR NEXT COMMAND *****

900  GOTO 100
```

```
1000  REM
****** INITIALIZATION
     ROUTINE ******

1005  REM
*** GET NUMBER OF ROOM
     S, ARTIFACTS, EFFECTS, AND M
     ONSTERS.
1010  PRINT DK$;"OPEN EAMON.DESC"
     : PRINT DK$;"READ EAMON.DESC
     ": INPUT NR,NZ,NE,NM: PRINT
     DK$;"CLOSE":NA = NZ
1019  REM
*** READ IN ARTIFACT D
     ATA.
1020  PRINT DK$;"OPEN EAMON.ARTIF
     ACTS.L128": DIM AN$(NA + 4),
     AD%(NA + 4,9): FOR A = 1 TO
     NA: PRINT DK$;"READ EAMON.AR
     TIFACTS,R";A: INPUT AN$(A): FOR
     A2 = 1 TO 4: INPUT AD%(A,A2)
     : NEXT A2: IF AD%(A,2) ) 1 THEN
      FOR A2 = 5 TO 8: INPUT AD%(
     A,A2): NEXT A2
1029  REM
*** READ IN MONSTER DA
     TA.
1030  NEXT A: PRINT DK$;"OPEN EAM
     ON.MONSTERS.L128": DIM MN$(N
     M),MD%(NM,15): FOR A = 1 TO
     NM: PRINT DK$;"READ EAMON.MO
     NSTERS,R";A: INPUT MN$(A): FOR
     A2 = 1 TO 12: INPUT MD%(A,A2
     ): NEXT A2,A: PRINT DK$;"CLO
     SE"
1039  REM
*** READ IN CHARACTER
     DATA
1040  PRINT DK$;"OPEN FRESH MEAT"
     : PRINT DK$;"READ FRESH MEAT
     ": INPUT REC: INPUT MN$(0),M
     D%(0,1),MD%(0,2),CH: FOR A =
     1 TO 4: INPUT SA%(A): NEXT :
      FOR A = 1 TO 5: INPUT WA%(A
     ): NEXT : INPUT AE,SEX$,GOLD
     ,BANK,AC
1049  REM
*** READ IN DATA FOR C
     HARACTER'S WEAPONS.
1050 NW = 15: DIM WN$(NW),WT%(NW)
     ,WO%(NW),WD%(NW),WP%(NW): FOR
     A = 1 TO 4: INPUT WN$(A),WT%
     (A),WO%(A),WD%(A),WS%(A): NEXT
     : PRINT DK$;"CLOSE"
1060  PRINT DK$;"OPEN EAMON.DESC,
     L256": PRINT DK$;"OPEN EAMON
     .ROOM NAMES,L64": PRINT DK$;
     "OPEN EAMON.ROOMS,L64"
1069  REM
*** PLAY WITH WEAPON N
     AMES TO ASSURE MONSTER WEAPO
     NS DON'T HAVE THE SAME NAMES
      AS CHARACTER WEAPONS.
1070  FOR W2 = 1 TO 4: IF WN$(W2)
      (   ) "NONE" THEN NA = NA +
     1:AN$(NA) = WN$(W2):AD%(NA,2
     ) = 2:AD%(NA,3) = 2:AD%(NA,4
     ) =  - 1:AD%(NA,5) = WO%(W2)
     :AD%(NA,6) = WT%(W2):AD%(NA,
     7) = WD%(W2):AD%(NA,8) = WS%
     (W2): NEXT :W2 = 4
1080 WZ = W2:WT = W * WZ
1090  FOR W2 = 1 TO WZ: FOR A = 1
      TO NA - WZ: IF AN$(A) = WN$
     (W2) THEN AN$(A) = AN$(A) +
     "#": FOR A = 0 TO 0: NEXT : GOTO
     1090
1100  NEXT A,W2:EA = 0:MD%(0,8) =
     0:A2 =  INT (AC / 2): IF A2 *
     2 (   ) AC THEN MD%(0,8) = 1:
     EA =  - 5
1110  IF A2 THEN MD%(0,8) = MD%(0
     ,8) + A2:EA = EA - A2 * 10: IF
     A2 = 3 THEN MD%(0,8) = MD%(0
     ,8) + 2:EA = EA - 30
1120  IF NA ) NZ THEN MD%(0,10) =
     (EA + AE) * ( - EA ) AE) + W
     O%(1) + WA%(WT%(1)) + 2 * MD
     %(0,2):MD%(0,7) = 0:MD%(0,9)
      = NZ + 1:MD%(0,11) = WD%(1)
     :MD%(0,12) = WS%(1):MD%(0,14
     ) = 3
1129  REM
*** READ COMMAND WORDS
      INTO AN ARRAY
1130  READ NC: DIM C$(NC): FOR C =
     1 TO NC: READ C$(C): NEXT
1140  FOR S = 1 TO 4:S2%(S) = SA%
     (S): NEXT : DIM V%(96):ROOM =
     1
1199  REM
*** READ IN ROOM NAMES
      AND DATA ON WHICH ROOM CONN
     ECTS WHERE.
1200  DIM ED%(NR,6)
1210  FOR X = 1 TO NR: PRINT DK$;
     "READ EAMON.ROOMS,R";X: FOR
     Y = 1 TO 6: INPUT ED%(X,Y): NEXT
     Y,X
1230  DIM RN$(NR)
1250  FOR X = 1 TO NR: PRINT DK$;
     "READ EAMON.ROOM NAMES,R";X:
      INPUT RN$(X): NEXT
1299  REM
*** RANDOM NUMBER FUNC
     TION TO RETURN INTEGER FROM
     1 TO 100.
1300  DEF  FN R(X) =  INT ( RND (
     1) * 100 + 1)
1349  REM
*** ERROR-HANDLING ROU
     TINE IS AT 59000
1350  ONERR  GOTO 59000
1899  REM
*** GOTO 100 TO START
     GAME.

1900  GOTO 100
1901  REM
*** DATA ***
1910  DATA     24
1920  DATA     NORTH,SOUTH,EAST,WE
     ST,UP,DOWN,GET,TAKE,DROP,LOO
     K,EXAMINE,ATTACK,FLEE,GIVE,I
     NVENTORY,BLAST,HEAL,POWER,SP
     EED,SMILE,WAVE,SAY,READY,SAV
     E
2000  REM
**** END GAME ROUTINE
     *****

2005  SPEED= 255
2010  PRINT : PRINT "(HIT ANY KEY
      TO CONTINUE)   ";: POKE  -
     16368,0: GET A$: PRINT
2020  PRINT DK$"CLOSE": IF SPD THEN
     MD%(0,2) = MD%(0,2) / 2
```

```
2029  REM
*** IF CHARACTER IS DE
     AD, SKIP THIS STUFF. ***
2030  IF DIE THEN 2500
2040  FOR W = 1 TO 4:WN$(W) = "NO
     NE": NEXT :W2 = 1: FOR A = 1
      TO NA: IF AD%(A,2) ) 1 AND
     AD%(A,4) =  - 1 THEN WN$(W2)
      = AN$(A):WP%(W2) = A:W2 = W
     2 + 1:AD%(A,4) = 0
2049  REM
*** CHECK FOR EXCESS W
     EAPONS.  MAKE HIM GET RID OF
      THEM. ***
2050  NEXT A: IF W2 ( 6 THEN 2300

2060  HOME : VTAB 5: PRINT "AS YO
     U START TO ENTER THE MAIN HA
     LL.": PRINT "LORD WILLIAM MI
     SSLEFIRE APPEARS AND": PRINT
     "TELLS YOU, 'YOU HAVE TOO MA
     NY WEAPONS": PRINT "TO KEEP
     THEM ALL--4 IS THE LEGAL LIM
     IT.": PRINT : PRINT "YOUR WE
     APONS ARE--"
2070  FOR W = 1 TO W2 - 1: PRINT
      SPC( 4);W;"--";WN$(W): NEXT
     : PRINT : PRINT "ENTER THE N
     UMBER OF THE WEAPON TO SELL
         (1-";W2 - 1;") "
2080  VTAB  PEEK (37): HTAB 23: INPUT
     A$:W5 =  VAL (A$): IF W5 ( 1
      OR W5 ) W2 - 1 OR W5 (  )  INT
     (W5) THEN 2080
2090 AD%(WP%(W5),4) =  - 1:WN$(W5
     ) = WN$(W2 - 1):WP%(W5) = WP
     %(W2 - 1):W2 = W2 - 1: IF W2
      ) 5 THEN 2060
2300  FOR W = 1 TO 4: IF WP%(W) THEN
     WT%(W) = AD%(WP%(W),6):WO%(W
     ) = AD%(WP%(W),5):WD%(W) = A
     D%(WP%(W),7):WS%(W) = AD%(WP
     %(W),8)
2309  REM
***** SELL TREASURE **
     ***

2310  NEXT W: HOME : VTAB 5: PRINT
     "AS YOU DELIVER YOUR TREASUR
     ES TO SAM": PRINT "SLICKER,
     THE LOCAL BUYER FOR SUCH": PRINT
     "THINGS, HE EXAMINES YOUR GO
     ODS AND": PRINT "PAYS YOU ";

2320 TP = 0: FOR A = 1 TO NA: IF
     AD%(A,4) (  )  - 1 THEN 2350

2330  IF AD%(A,2) (  ) 0 THEN AD%
     (A,1) = AD%(A,1) * CH / 10
2340 TP = TP + AD%(A,1)
2350  NEXT A: PRINT TP;" GOLD PIE
     CES.": PRINT
2360 GOLD = GOLD + TP: IF GOLD (
     0 THEN GOLD = 0
2370  PRINT : PRINT "(HIT ANY KEY
      TO CONTINUE)  ";: GET A$: PRINT

2499  REM
***** GO HOME *****

2500  ONERR  GOTO 2500: REM *** R
     ESTART HERE ON ERROR
2510  HOME : VTAB 5: PRINT "(INSE
     RT EAMON MASTER DISKETTE, TH
     EN": PRINT "   HIT THE 'C' K
     EY)  ";: POKE  - 16368,0
2520  GET A$: IF A$ (  ) "C" THEN
     2520
2530  PRINT A$
```

```
2540  IF DIE THEN  PRINT DK$;"OPE
     N THE ADVENTURER,S6,D1": PRINT
     DK$;"DELETE THE ADVENTURER":
      GOTO 2900
2549  REM
*** WRITE CURRENT CHAR
     ACTER STATISTICS TO MAIN CHA
     RACTER FILE ***
2550  PRINT DK$;"OPEN CHARACTERS,
     L150,S6,D1": PRINT DK$;"WRIT
     E CHARACTERS,R";REC: PRINT M
     N$(0): PRINT MD%(0,1): PRINT
     MD%(0,2): PRINT CH: FOR A =
     1 TO 4: PRINT SA%(A): NEXT
2560  FOR A = 1 TO 5: PRINT WA%(A
     ): NEXT : PRINT AE: PRINT SE
     X$: PRINT GOLD: PRINT BANK: PRINT
     AC: FOR A = 1 TO 4: PRINT WN
     $(A): PRINT WT%(A): PRINT WO
     %(A): PRINT WD%(A): PRINT WS
     %(A): NEXT
2569  REM
*** WRITE POINTER TO C
     HARACTER FILE ***
2570  PRINT DK$;"OPEN THE ADVENTU
     RER": PRINT DK$;"WRITE THE A
     DVENTURER": PRINT MN$(0): PRINT
     REC: PRINT DK$;"CLOSE"
2899  REM
***** RUN MAIN HALL PR
     OGRAM ON MASTER DISK *****

2900  PRINT DK$;"RUN MAIN HALL,S6
     ,D1"
3000  REM
***** MOVE COMMAND ***
     **

3010  IF NBTL AND S$ (  ) "FLEE" THEN
      PRINT : PRINT "YOU CAN'T DO
      THAT WITH UNFRIENDLIES
        ABOUT!": PRINT : GOTO 100

3019  REM
*** FIND DIRECTION PLA
     YER WANTS TO MOVE
3020  FOR D = 1 TO 6: IF  LEFT$ (
     V$,1) (  )  MID$ ("NSEWUD",D
     ,1) THEN  NEXT
3030 R2 = ED%(ROOM,D)
3039  REM
*** IF R2 IS ZERO, YOU
      CAN'T GO THAT WAY. IF R2 IS
      NEGATIVE, MOVING THAT DIREC
     TION IS A 'SPECIAL MOVE.'
3040  IF R2 ) 0 THEN 3500
3050  REM
*** SPECIAL MOVES.
3059  REM
*** GOING TO ROOM -99
     TAKES YOU HOME.
3060  IF R2 =  - 99 THEN  PRINT :
      PRINT "YOU SUCCESSFULLY RID
     E OFF INTO THE": PRINT "  SU
     NSET.": GOTO 2000
3070  IF  NOT R2 THEN 3490
3490  PRINT : PRINT "YOU CAN'T GO
      THAT WAY!":V%(ROOM) = 0: GOTO
     100
3499  REM
*** SUCCESSFUL MOVE. R
     2 IS NEW ROOM. R3 IS OLD ROO
     M.
3500 R3 = ROOM:ROOM = R2: GOSUB 3
     600
3590  GOTO 100
```

```
3600  REM
*** CHECK MONSTER REAC
    TION: FRIENDLY, NEUTRAL, OR
    HOSTILE.

3610 TD%(1) = 0:TD%(3) = MD%(0,1)
    :FD%(1) = 0:FD%(3) = MD%(0,1
    3)
3619  REM
*** DID MONSTER FOLLOW
     CHARACTER INTO ROOM? ***
3620  FOR M = 1 TO NM: IF MD%(M,5
    ) = R3 THEN  IF MD%(M,14) =
    3 OR (MD%(M,14) = 1 AND 200 *
     RND (1) < MD%(M,4)) THEN MD
    %(M,5) = ROOM: GOTO 3670
3630  IF MD%(M,5) <  > ROOM THEN
    3900
3639  REM
*** IF MONSTER HAS BEE
    N MET BEFORE, SKIP NEXT SECT
    ION. ***
3640  IF MD%(M,14) THEN 3670
3649  REM
*** FIND REACTION OF M
    ONSTER ***
3650 FR = MD%(M,3): IF FR AND FR <
     > 100 THEN FR = FR +  INT (
    (CH - 10) / 2)
3660 MD%(M,14) = 1: IF FR > (100 *
     RND (1)) THEN MD%(M,14) = 2
    : IF FR > (100 *  RND (1)) THEN
    MD%(M,14) = 3
3670 A = MD%(M,14):FD%(A) = FD%(A
    ) + MD%(M,13):TD%(A) = TD%(A
    ) + MD%(M,1)
3895  REM
*** LOOP UNTIL WE RUN
    OUT OF MONSTERS ***
3900  NEXT M:NBTL = (TD%(1) > 0):
     RETURN
4000  REM
***** GET COMMAND ****
    *


4010  GOSUB 4900
4020  IF S$ = "ALL" THEN 4160
4100  REM
*** IS ITEM HERE? ***
4120  FOR A = 1 TO NA: IF (AN$(A)
    <  > S$ AND  LEFT$ (AN$(A),
    LEN (S$)) <  > S$ AND  RIGHT$
    (AN$(A), LEN (S$)) <  > S$) OR
    AD%(A,4) <  > ROOM THEN  NEXT
    : PRINT : PRINT "I SEE NO ";
    S$;" HERE!": PRINT : GOTO 10
    0
4129  REM
*** IS IT TOO HEAVY? *
    **
4130  GOSUB 4200: IF WT + AD%(A,3
    ) > 10 * MD%(0,1) THEN  PRINT
    : PRINT "IT IS TOO HEAVY FOR
     YOU.": PRINT : GOTO 100
4139  REM
*** GOT IT. ***
4140  PRINT : PRINT "GOT IT.":AD%
    (A,4) =  - 1:WT = WT + AD%(A
    ,3): PRINT : IF AD%(A,2) < 2
     OR MD%(0,9) <  >  - 1 THEN
    300
4149  REM
*** IF ITEM IS A WEAPO
    N, AND CHARACTER IS UNARMED,
     READY THE WEAPON. ***
```

```
4150  GOTO 17000
4159  REM
*** GET ALL COMMAND **

4160  FOR A = 1 TO NA: IF AD%(A,4
    ) <  > ROOM THEN 4190
4170  GOSUB 4200: IF WT + AD%(A,3
    ) > 10 * MD%(0,1) THEN  PRINT
    AN$(A);" IS TOO HEAVY": GOTO
    4190
4180  PRINT AN$(A) + " TAKEN.":AD
    %(A,4) =  - 1:WT = WT + AD%(
    A,3)
4190  NEXT : PRINT : GOTO 300
4200  REM
*** INSERT SPECIAL EFF
    ECTS OF PICKING SOMETHING UP
     HERE ***
4210  RETURN
4900  REM
*** ROUTINE TO GET S$
    IF NOT PREVIOUSLY SPECIFIED.
     ***
4905  IF S$ = "" THEN  PRINT : PRINT
    C$(C);" WHAT";: INPUT S$: GOTO
    4900
4910  RETURN
5000  REM
***** DROP COMMAND ***
    **


5010  GOSUB 4900
5020  IF S$ = "ALL" THEN 5100
5029  REM
*** DOES HE HAVE IT? *
    **
5030  FOR A = 1 TO NA: IF (AN$(A)
     <  > S$ AND  LEFT$ (AN$(A),
     LEN (S$)) <  > S$ AND  RIGHT$
    (AN$(A), LEN (S$)) <  > S$) OR
    AD%(A,4) <  >  - 1 THEN  NEXT
    : PRINT : PRINT "YOU AREN'T
    CARRYING A ";S$: PRINT : GOTO
    100
5039  REM
*** DROP IT. ***
5040  WT = WT - AD%(A,3):AD%(A,4) =
    ROOM: PRINT AN$(A);" DROPPED
    .": PRINT
5049  REM
*** TOO MUCH WEIGHT RO
    UTINE ***
5050  IF WT > 10 * MD%(0,1) THEN
     PRINT "YOU SUDDENLY FIND YO
    U CANNOT CARRY ALL  OF THE I
    TEMS YOU ARE CARRYING, AND T
    HEY ALL FALL TO THE GROUND."
    : PRINT : GOTO 5100
5060  IF MD%(0,9) > 0 THEN  IF AD
    %(MD%(0,9),4) <  >  - 1 THEN
    MD%(0,9) =  - 1
5070  GOTO 300
5099  REM
*** DROP ALL ROUTINE *
    **
5100  FOR A = 1 TO NA: IF AD%(A,4
    ) =  - 1 THEN  PRINT AN$(A);
    " DROPPED.":AD%(A,4) = ROOM
5110  NEXT A:WT = 0: GOTO 5060
6000  REM
***** LOOK COMMAND ***
    **
```

```
6019  REM
*** GIVE ARTIFACT DESC
     RIPTION ***
6020 LK = 0: FOR A = 1 TO NZ: IF
     AN$(A) = S$ AND (AD%(A,4) =
     ROOM OR AD%(A,4) =  - 1) THEN
      PRINT DK$;"READ EAMON.DESC.
     R";A + 100: INPUT A$: PRINT
     DK$: PRINT : PRINT A$: PRINT
     :LK = 1
6030  NEXT A: IF LK THEN 300
6039  REM
*** GIVE MONSTER DESCR
     IPTION ***
6040  FOR A = 1 TO NM: IF MN$(A) =
     S$ AND MD%(A,5) = ROOM THEN
      PRINT DK$;"READ EAMON.DESC,
     R";A + 300: INPUT A$: PRINT
     DK$: PRINT : PRINT A$: PRINT
     :LK = 1
6050  NEXT A: IF LK THEN 300
6059  REM
*** SET FLAG SO ROOM D
     ESCRIPTION WILL BE GIVEN IN
     MAIN LOOP ***
6060 V%(ROOM) = 0: GOTO 300
7000  REM
*********** ATTACK COMM
     AND **********

7010  GOSUB 4900
7279  REM
*** MAKE SURE VICTIM I
     S HERE. ***
7280  FOR M = 1 TO NM: IF (S$ (  )
     MN$(M) AND  LEFT$ (MN$(M), LEN
     (S$)) (  ) S$ AND  RIGHT$ (M
     N$(M), LEN (S$)) (  ) S$) OR
     MD%(M,5) (  ) ROOM THEN  NEXT
      : PRINT : PRINT "ATTACK WHO?
     ": PRINT : GOTO 100
7289  REM
*** IS ATTACKER ARMED?
      ***
7290  IF MD%(0,9) =  - 1 THEN  PRINT
      : PRINT "YOU HAVE NO WEAPON
     READY!": PRINT : GOTO 100
7299  REM
**** DO THE ATTACK. IF
      IT DOESN'T WORK, SKIP.****
7300 OF = 0:DF = M: GOSUB 7500: IF
      NOT HIT THEN 7360
7309  REM
*** CHECK FOR SKILL IN
     CREASE. ***
7310  IF INC THEN W2 = AD%(MD%(0,
     9),6):WA%(W2) = WA%(W2) + 2
7319  REM
*** CHECK FOR INCREASE
      IN ARMOR EXPERTISE. ***
7320  IF  - EA (  = AE THEN 300
7330  IF  INT (100 *  RND (1) + 1
     ) (  = MD%(0,10) THEN 7360
7340 A = 2: IF EA + AE =  - 1 THEN
     A = 1
7350 AE = AE + A:MD%(0,10) = MD%(
     0,10) + A
7360  IF MD%(DF,14) = 1 OR DF = 0
      THEN 300
7369  REM
*** ATTACKING A MONSTE
     R DECREASES ITS FRIENDLINESS
       ***
7370 MD%(DF,3) = MD%(DF,3) / 2: FOR
     M = 1 TO NM: IF MD%(M,5) = R
     OOM AND MD%(M,14) ) 1 THEN M
     D%(M,14) = 0
7380  NEXT M:R3 = ROOM: GOSUB 360
     0: GOTO 300
```

```
7400  REM
**** MONSTER TRIES TO
     PICK UP A WEAPON ****
7410  IF MD%(OF,10) = 0 THEN  RETURN

7419  REM
*** SEE IF THERE'S A W
     EAPON ON THE FLOOR ***
7420  FOR A = 1 TO NA: IF AD%(A,4
     ) (  ) ROOM OR AD%(A,2) ( 2 THEN
      NEXT : RETURN
7430  PRINT MN$(OF);" PICKS UP ";
     AN$(A): PRINT :AD%(A,4) = 0:
     MD%(OF,9) = A:MD%(OF,10) = M
     D%(OF,10) + AD%(A,5):MD%(OF,
     4) = MD%(OF,4) * 2: RETURN
7499  REM
***** ATTACK ROUTINE *
     ****

7500 HIT = 0: IF MD%(OF,9) =  - 1
      THEN 7400
7504  REM
*** PRINT ATTACK MESSA
     GE AND ROLL THE 'DICE'. ***
7505  PRINT MN$(OF);" ATTACKS ";M
     N$(DF): PRINT " --";:RL =  INT
     (100 *  RND (1) + 1):HIT = 0
     : IF (RL ( 5 OR RL ( MD%(OF,
     10) - MD%(DF,7)) AND RL ( 96
      THEN HIT = 1
7510  IF HIT THEN 7600
7519  REM
*** A MISS.  IS IT A F
     UMBLE? ***
7520  IF RL ( 97 THEN  PRINT "A M
     ISS.": RETURN
7525  INVERSE : PRINT "A FUMBLE!"
     : NORMAL :FR =  FN R(X)
7527  IF FR ( 35 + 40 * ( NOT MD%
     (OF,9)) THEN  PRINT "   FUMB
     LE RECOVERED.": PRINT : RETURN

7530  IF FR ( 76 THEN  PRINT "  W
     EAPON DROPPED!":AD%(MD%(OF,9
     ),4) = ROOM:MD%(OF,10) = MD%
     (OF,10) - AD%(MD%(OF,9),5):M
     D%(OF,9) =  - 1:MD%(OF,4) =
     MD%(OF,4) / 2: PRINT : RETURN

7532  IF MD%(OF,9) = 0 GOTO 7550
7535  IF FR ) 95 THEN 7550
7536  REM
*** HIGH-QUALITY WEAPO
     NS BREAK LESS OFTEN. ***
7537  IF AD%(MD%(OF,9),5) * 3 (  FN
     R(X) THEN  PRINT "  FUMBLE
     RECOVERED.": PRINT : RETURN

7540  PRINT "  WEAPON BROKEN!":AD
     %(MD%(OF,9),4) = 0:MD%(OF,10
     ) = MD%(OF,10) - AD%(MD%(OF,
     9),5):MD%(OF,9) =  - 1:MD%(O
     F,4) = MD%(OF,4) / 2: IF  RND
     (1) ) .5 THEN  PRINT : RETURN

7545  PRINT "  BROKEN WEAPON HURT
     S USER!"
7550  IF FR = 100 THEN 7560
7555 D = MD%(OF,11):S = MD%(OF,12
     ):DF = OF:A = 1: GOTO 7635
7560 D = MD%(OF,11) * 2:S = MD%(O
     F,12):DF = OF:A = 0: GOTO 76
     35
7599  REM
*** A HIT!  CHECK FOR
     SKILL INCREASE ***
```

```
7600 INC = ( INT (100 *  RND (1) +
     1) ) MD%(OF,10)):MD%(OF,10) =
     MD%(OF,10) + 2 * INC
7605 D = MD%(OF,11):S = MD%(OF,12
     ):A = 1: IF RL ) 5 THEN  PRINT
     "A HIT!": GOTO 7635
7609  REM
*** CRITICAL HIT ROUTI
     NE.  CALCULATE EXTRA DAMAGE
     ***
7610  INVERSE : PRINT "A CRITICAL
      HIT!": NORMAL :R2 =  FN R(X
     ): IF R2 ( 51 THEN A = 0: GOTO
     7635
7615  IF R2 ( 86 THEN S = 1.5 * S
     : GOTO 7635
7620  IF R2 ( 96 THEN D = 2 * D: GOTO
     7635
7625  IF R2 ( 100 THEN D = 3 * D:
      GOTO 7635
7630  GOTO 7700
7634  REM
*** CALCULATE DAMAGE.
     SUBTRACT ARMOR VALUE FROM DA
     MAGE. ***
7635 D2 = 0: FOR D3 = 1 TO D:D2 =
     D2 +  INT (S *  RND (1) + 1)
     : NEXT :D2 = D2 - A * MD%(DF
     ,8): IF D2 ( 1 THEN  PRINT "
        BLOW BOUNCES OFF ARMOUR":
      PRINT : RETURN
7640 MD%(DF,13) = MD%(DF,13) + D2
     : IF MD%(DF,13) )  = MD%(DF,
     1) THEN 7700
7645 A = MD%(DF,14):FD%(A) = FD%(
     A) + D2
7649  REM
*** TELL PLAYER HOW HE
     ALTHY THE DEFENDER IS. ***
7650  PRINT : PRINT MN$(DF);: ON
      INT (MD%(DF,13) * 5 / MD%(D
     F,1) + 1) GOTO 7655,7660,766
     5,7670,7675
7655  PRINT " IS IN GOOD SHAPE.":
      RETURN
7660  PRINT " IS LIGHTLY INJURED.
     ": RETURN
7665  PRINT " IS BADLY INJURED.":
      RETURN
7670  PRINT " IS VERY BADLY INJUR
     ED.": RETURN
7675  PRINT " IS AT DEATH'S DOOR.
     ": PRINT "   KNOCKING LOUDLY
     .": RETURN
7700  FLASH : PRINT MN$(DF);" IS
     DEAD!": NORMAL
7702 A = MD%(DF,14):FD%(A) = FD%(
     A) + MD%(DF,1) - MD%(DF,13) +
     D2:MD%(DF,5) = 0
7703  REM
*** PUT THE DEAD BODY
     IN THE ROOM. ***
7704 A2 = DF + NZ - NM:AD%(A2,4) =
     ROOM
7705  REM
*** DROP WEAPONS, DISA
     RM, REDUCE COURAGE (IN CASE
     MONSTER GETS RESURRECTED).
7706  IF MD%(DF,9) ) 0 THEN AD%(M
     D%(DF,9),4) = ROOM:MD%(DF,10
     ) = MD%(DF,10) - AD%(MD%(DF,
     9),5):MD%(DF,9) =  - 1:MD%(D
     F,4) = MD%(DF,4) / 2
```

```
7709  REM
*** IF DEAD CHARACTER
     IS PLAYER CHARACTER, END GAM
     E ***
7710 A2 = DF + NZ - NM:AD%(A2,4) =
     ROOM: FOR AJ = 1 TO NA: IF A
     D%(AJ,4) =  - DF - 1 THEN AD
     %(AJ,4) = ROOM
7720  NEXT : IF MD%(DF,9) ) 0 THEN
     AD%(MD%(DF,9),4) = ROOM:MD%(
     DF,10) = MD%(DF,10) - AD%(MD
     %(DF,9),5):MD%(DF,9) =  - 1:
     MD%(DF,4) = MD%(DF,4) / 2
7740  IF DF = 0 THEN DIE = 1: GOTO
     2000
7900  PRINT : RETURN
8000  REM
***** FLEE COMMAND ***
     **

8010  IF  NOT NBTL THEN  PRINT : PRINT
     "THERE'S NOTHING TO FLEE FRO
     M!": PRINT : GOTO 100
8020  FOR D = 1 TO 6: IF S$ = C$(
     D) THEN V$ = S$:S$ = "FLEE":
      GOTO 3000
8030  NEXT
8399  REM
*** FIND A PLACE TO FL
     EE TO ***
8490 DR$ = "": GOSUB 8500: IF DR$
      = "000000" THEN  PRINT : PRINT
     "THERE'S NO PLACE TO GO!!": GOTO
     100
8495  GOTO 3500
8500 K1 = 0: FOR D = 1 TO 6:K = E
     D%(ROOM,D):DR$ = DR$ +  STR$
     (K): IF K ) 0 THEN K1 = K1 +
     1:DR%(K1) = K
8510  NEXT :RR =  RND (1) * K1 +
     1:R2 = DR%(RR)
8520  RETURN
9000  REM
***** GIVE COMMAND. **
     ***

9010 A2 = 1
9019  REM
*** SEE IF S$ CONTAIN
     S A MONSTER NAME ***
9020  FOR M = 1 TO NM: IF  LEFT$
     (S$, LEN (MN$(M))) = MN$(M) AND
     MD%(M,5) = ROOM THEN 9050
9029  REM
*** IF NOT, GET MONSTE
     R NAME ***
9030  NEXT : IF A2 = 1 THEN A2 =
     2: PRINT : INPUT "WHO TO GIV
     E TO? ";S$: GOTO 9020
9040  PRINT : PRINT "NOBODY HERE
     BY THAT NAME!": PRINT : GOTO
     100
9050 S$ =  MID$ (S$, LEN (MN$(M))
      + 1)
9060  IF  LEFT$ (S$,1) = " " THEN
     S$ =  MID$ (S$,2): GOTO 9060

9064  REM
*** IF VALUE OF S$ )0,
      S$ IS A NUMBER. ASSUME PLAY
     ER WANTS TO GIVE GOLD PIECES
       ***
9065  IF  VAL (S$) ) 0 THEN 9500
```

```
9069  REM
*** SEE IF S$ CONTAINS
     AN ARTIFACT NAME ***
9070  FOR A = 1 TO NA: IF  LEFT$
     (S$, LEN (AN$(A))) = AN$(A) AND
     AD%(A,4) =  - 1 THEN 9095
9079  REM
*** IF NOT, GET AN ART
     IFACT NAME ***
9080  NEXT : IF A2 = 2 THEN A2 =
     3: PRINT : INPUT "WHAT TO GI
     VE? ";S$: GOTO 9065
9090  PRINT : PRINT "YOU AREN'T C
     ARRYING A ";S$;"!": PRINT : GOTO
     100
9095  IF A = MD%(0,9) THEN MD%(0,
     9) =  - 1
9100  PRINT : PRINT "OKAY.":AD%(A
     ,4) =  - M - 1:MD%(M,14) = 0
     :MD%(M,3) = MD%(M,3) * (1 +
     AD%(A,1) / 100)
9110  IF MD%(M,10) (  ) 0 AND MD%
     (M,9) =  - 1 THEN MD%(M,9) =
     A:MD%(M,10) = MD%(M,9) + AD%
     (A,5):MD%(M,4) = 2 * MD%(M,4
     )
9120 R3 = ROOM: GOSUB 3600: GOTO
     5050
9499  REM
*** GIVE GOLD ***
9500  IF  VAL (S$) ) GOLD THEN  PRINT
     : PRINT "YOU AREN'T CARRYING
      THAT MUCH GOLD OF       YOUR
      OWN!": PRINT : GOTO 300
9510 GOLD =  INT (GOLD -  VAL (S$
     )):MD%(M,14) = 0:MD%(M,3) =
     MD%(M,3) * (1 +  VAL (S$) /
     100): GOTO 9120
10000  REM
***** INVENTORY COMMA
     ND *****

10010  PRINT : PRINT "YOU ARE CAR
     RYING THE FOLLOWING--":A0 =
     0: FOR A = 1 TO NA: IF AD%(A
     ,4) =  - 1 THEN  PRINT " ";A
     N$(A); LEFT$ (" (READY WEAPO
     N)",1 + 20 * (A = MD%(0,9)))
     :A0 = A0 + 1: IF A0 = 15 THEN
      PRINT : PRINT "(HIT ANY KEY
      TO CONTINUE)";: GET A$: PRINT
     :A0 = 0
10020  NEXT : GOTO 300
11000  REM
***** BLAST SPELL ***
     **


11010  GOSUB 4900
11469  REM
*** SEE IF TARGET IS
     BLASTABLE ***
11470  FOR M = 1 TO NM: IF S$ (  )
     MN$(M) OR MD%(M,5) (  ) ROOM
      THEN  NEXT : PRINT : PRINT
     "YOU CAN'T BLAST ";S$;"!": PRINT
     : GOTO 100
11479  REM
*** ATTEMPT TO BLAST
     THE TARGET. SUC = SUCCESS FL
     AG ***
11480 S = 1: GOSUB 11500: IF  NOT
     SUC THEN 300
11489  REM
*** SUCCESSFUL BLAST.
      ***
11490  PRINT : PRINT "DIRECT HIT!
     ": PRINT :S = 6:D = 1:DF = M
     :A = 0: GOSUB 7635: PRINT : GOTO
     300
```

```
11499  REM
*** BLAST SUBROUTINE.
      SEE IF SPELL IS SUCCESSFUL
      ***
11500 SUC = 0: IF SA%(S) THEN RL =
     FN R(X):SUC = (((RL ( S2%(S
     )) OR (RL ( 6)) AND (RL ( 96
     ))
11510  IF SUC THEN 11520
11511  REM
*** IF ROLL=1 THEN SP
     ELL HAS BEEN FORGOTTEN. ***
11512  IF  FN R(X) = 1 THEN  GOSUB
     11600: RETURN
11513  REM
*** FAILURE. ***
11514  PRINT "  NOTHING HAPPENS."
     RETURN
11519  REM
*** TEST FOR SKILL IN
     CREASE. ***
11520 RL =  INT (100 *  RND (1) +
     1): IF RL ) S2%(S) THEN S2%(
     S) = S2%(S) + 2:SA%(S) = SA%
     (S) + 2
11530 S2%(S) = S2%(S) / 2
11540  RETURN
11599  REM
*** FORGET SPELL SUBR
     OUTINE. ***
11600  PRINT : PRINT "THE STRAIN
     OF ATTEMPTING TO CAST THIS
         SPELL OVERLOADS YOUR BRA
     IN AND YOU      FORGET IT CO
     MPLETELY.": PRINT :SA%(S) =
     0: RETURN
12000  REM
***** HEAL SPELL ****
     *


12010 S = 2: GOSUB 11500: IF  NOT
     SUC THEN 300
12020  PRINT : IF MD%(0,13) THEN
      PRINT "SOME OF YOUR WOUNDS
     SEEM TO CLEAR UP.": PRINT
12030 MD%(0,13) = MD%(0,13) -  INT
     (10 *  RND (1) + 1): IF MD%(
     0,13) ( 0 THEN MD%(0,13) = 0

12040 DF = 0: GOSUB 7650: GOTO 30
     0
13000  REM
***** POWER SPELL --
     RE-WRITE FOR SPECIAL EFFECTS
      *****


13005  REM
*** RANDOM EFFECTS **
     *
13010 S = 4: GOSUB 11500: IF  NOT
     SUC THEN 300
13019  REM
*** RESURRECTION OF A
      DEAD MONSTER? ***
13020 RAISE = 0: FOR M = 1 TO NM:
     M2 = NZ - NM + M: IF AD%(M2,
     4) = ROOM OR AD%(M2,4) =  -
     1 THEN RAISE = 1: PRINT MN$(
     M);" COMES ALIVE!":AD%(M2,4)
      = 0:MD%(M,5) = ROOM:MD%(M,1
     3) = 0:WT = WT - AD%(M2,3)
13030  NEXT M: IF RAISE THEN R3 =
     ROOM: GOSUB 3600: GOTO 300
13039  REM
*** AN ARTIFACT VANIS
     HES? ***
```

```
13040  FOR A = 1 TO NA: IF AD%(A,
      4) = ROOM THEN RAISE = 1: PRINT
      AN$(A):" VANISHES!":AD%(A,4)
       = 0
13050  NEXT A: IF RAISE THEN  PRINT
      : GOTO 300
13059  REM
*** ROOF COLLAPSES? *
      **
13060 RR =  INT (100 *  RND (1) +
      1): IF RR < 11 THEN  PRINT :
       PRINT "THE SECTION OF TUNNE
      L YOU ARE IN ": PRINT "  COL
      LAPSES AND YOU DIE.":DIE = 1
      : GOTO 2000
13079  REM
*** SONIC BOOM? ***
13080  IF RR < 86 THEN  PRINT : PRINT
      "YOU HEAR A VERY LOUD SONIC
      BOOM": PRINT "   THAT ECHOES
       THROUGH THE TUNNELS.": PRINT
      : GOTO 300
13089  REM
*** HEALING? ***
13090  IF RR > 95 THEN  PRINT : PRINT
      "ALL OF YOUR HITS ARE HEALED
      .": PRINT :MD%(0,13) = 0: GOTO
      300
13100  GOTO 14020
14000  REM
***** SPEED SPELL: IN
      CREASE AGILITY AND CHANCE TO
       HIT *****

14010 S = 3: GOSUB 11500: IF  NOT
      SUC THEN 300
14020  IF  NOT SPD THEN MD%(0,10)
       = MD%(0,10) + 2 * MD%(0,2):
      MD%(0,2) = 2 * MD%(0,2)
14030 SPD = SPD +  INT (25 *  RND
      (1) + 10): PRINT : PRINT "YO
      U CAN FEEL THE NEW AGILITY F
      LOWING": PRINT "   THROUGH Y
      OU!": PRINT : GOTO 300
15000  REM
***** SMILE COMMAND:
      TRY TO MAKE FRIENDS *****

15010  PRINT : FOR M = 1 TO NM: IF
      MD%(M,5) <  > ROOM THEN 1510
      0
15020  ON MD%(M,14) GOTO 15030,15
      040,15050
15030  PRINT MN$(M);" GROWLS AT Y
      OU": GOTO 15100
15040  PRINT MN$(M);" IGNORES YOU
      ": GOTO 15100
15050  PRINT MN$(M);" ";V$;"S BAC
      K": GOTO 15100
15100  NEXT M: PRINT : GOTO 300
16000  REM
**** SAY COMMAND ****
      *

16010  GOSUB 4900
16030  IF S$ = "BLAST" THEN V$ =
      S$:S$ = "": GOTO 11000
16040  IF S$ = "HEAL" THEN 12000
16050  IF S$ = "POWER" THEN 13000

16060  IF S$ = "SPEED" THEN 14000

16900  PRINT : PRINT "OKAY, '";S$
      ;"'": PRINT : GOTO 300
```

```
17000  REM
***** READY A WEAPON.
      *****

17010  GOSUB 4900
17019  REM
*** IS HE CARRYING TH
      E WEAPON? ***
17020  FOR A = 1 TO NA: IF AN$(A)
       <  > S$ OR AD%(A,4) <  > -
      1 THEN  NEXT : PRINT : PRINT
      "YOU AREN'T CARRYING IT!": PRINT
      : GOTO 100
17029  REM
*** IS IT REALLY A WE
      APON? ***
17030  IF AD%(A,2) < 2 THEN  PRINT
      : PRINT "THAT ISN'T A WEAPON
      !": PRINT : GOTO 100
17979  REM
*** READY IT. ***
17980 MD%(0,10) = (EA + AE) * ( -
      EA > AE) + AD%(A,5) + WA%(AD
      %(A,6)) + 2 * MD%(0,2):MD%(0
      ,9) = A:MD%(0,11) = AD%(A,7)
      :MD%(0,12) = AD%(A,8)
17990  PRINT : PRINT "READIED.": PRINT
      : GOTO 300
18000  REM
*** SAVE GAME ***

18010  PRINT : PRINT "DO YOU WANT
       TO SAVE THIS GAME? ": PRINT
      "(Y/N) :";
18020  GET A$: IF A$ <  > "Y" AND
      A$ <  > "N" THEN 18020
18030  PRINT A$: IF A$ = "N" THEN
      100
18035 DK$ =  CHR$ (4)
18040  PRINT DK$;"CLOSE":X =  FRE
      (0)
18050  PRINT DK$"BSAVE GAME.PTRS,
      A$69,L8"
18060  PRINT DK$"BSAVE GAME.SVAR,
      A"; PEEK (105) +  PEEK (106)
       * 256;",L"; PEEK (109) +  PEEK
      (110) * 256 -  PEEK (105) -
       PEEK (106) * 256 + 1
18070  PRINT DK$"BSAVE GAME.STR,A
      "; PEEK (111) +  PEEK (112) *
      256;",L"; PEEK (115) +  PEEK
      (116) * 256 -  PEEK (111) -
       PEEK (112) * 256 + 1
18080  PRINT : PRINT "THE GAME IS
       NOW SAVED. BOOT ON THIS": PRINT
      "  DISKETTE TO RESTART THE G
      AME."
18090  END
19000  REM
*** RESTART OLD GAME *
      **

19010  PRINT : PRINT "DO YOU WANT
       TO RESTART THIS GAME?": PRINT
      "  Y OR N:";
19020  GET A$: IF A$ <  > "Y" AND
      A$ <  > "N" THEN 19020
19030  PRINT A$: IF A$ = "N" THEN
       END
19035 DK$ =  CHR$ (4)
19040  PRINT DK$"BLOAD GAME.PTRS"
      : PRINT DK$"BLOAD GAME.SVAR"
      : PRINT DK$"BLOAD GAME.STR"
```

```
19050  PRINT DK$;"DELETE GAME.PTR
     S": PRINT DK$;"DELETE GAME.S
     VAR": PRINT DK$;"DELETE GAME
     .STR"
19060  PRINT DK$;"OPEN EAMON.DESC
     ,L256": PRINT DK$;"OPEN EAMO
     N.ROOMS,L64": PRINT DK$;"OPE
     N EAMON.ROOM NAMES,L64"
19070  GOTO 100
50000  REM
***** ERROR HANDLING
     ROUTINE *****
59000  REM
*** FIRST POKE IN A R
     OUTINE TO HELP DEFEAT APPLES
     OFT'S 'ONERR GOTO' BUGS. ***
59001  POKE 768,104: POKE 769,168
     : POKE 770,104: POKE 771,166
     : POKE 772,223: POKE 773,154
     : POKE 774,72: POKE 775,152:
      POKE 776,72: POKE 777,96: CALL
     768
59004 X =  FRE (0): REM ** THIS A
     LSO HELPS KILL BUGS.
59005  NORMAL
59006  REM
*** PRINT ERROR #, LINE # ***

59010 CODE =  PEEK (222):LN =  PEEK
     (218) + 256 *  PEEK (219)
59020  PRINT  CHR$ (7);"**** ERRO
     R #";CODE;" IN LINE ";LN
59030  IF CODE = 255 THEN  STOP :
      REM  ** BREAK ATTEMPTED
59199  PRINT
59200  PRINT "*** ATTEMPTING TO R
     ECOVER ***": PRINT : GOTO 10
     0
59999  END
60000  REM
*** THAT'S ALL, FOLKS! ***
```